Managing the Java Thread Lifecycle: Layers Involved in Starting a Thread

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Lesson

• Understand the layers involved in starting a Java thread
Layers Involved in Starting a Java Thread

• Starting a Java thread involves interesting design & implementation issues
Layers Involved in Starting a Java Thread
Layers Involved in Starting a Java Thread

- Calling start() on a thread triggers the execution of its run() hook method.

- New (`MyThread()`)

- Runnable

- Running

- Terminated

- Waited

- Timed Waiting

- Blocked

- attempt to access guarded resource

- cond.notify(), cond.notifyAll()

- cond.wait()

- run() method returns

- myThread.sleep() wait(timeout) join(timeout)
Layers Involved in Starting a Java Thread

- The Java platform provides a stack of layers that define various mechanisms for running concurrent programs on a wide range of computing devices.

Different versions of Android & Java implement these layers differently, though key levels of abstraction are often similar.

• Likewise, the Android platform provides a stack of layers that define various mechanisms for running concurrent programs on mobile computing devices.
Layers Involved in Starting a Java Thread

- Likewise, the Android platform provides a stack of layers that define various mechanisms for running concurrent programs on mobile computing devices.
Layers Involved in Starting a Java Thread

- Likewise, the Android platform provides a stack of layers that define various mechanisms for running concurrent programs on mobile computing devices.

The Bionic LibC library supports the Pthreads C programming APIs.
Layers Involved in Starting a Java Thread

- Likewise, the Android platform provides a stack of layers that define various mechanisms for running concurrent programs on mobile computing devices.

Dalvik & ART provide a managed execution environment for Java apps.
Layers Involved in Starting a Java Thread

- Likewise, the Android platform provides a stack of layers that define various mechanisms for running concurrent programs on mobile computing devices.

Package java.util.concurrent

Description

Utility classes commonly useful in concurrent programming. This package includes a few small standardized extensible frameworks, as well as some classes that provide useful functionality and are otherwise tedious or difficult to implement. Here are brief descriptions of the main components. See also the java.util.concurrent.locks and java.util.concurrent.atomic packages.

Android’s runtime contains the classes in the java.util.concurrent packages
Layers Involved in Starting a Java Thread

- Creating & starting new threads on any Java platform consumes a non-trivial amount of system resources, so use them judiciously!
Layers Involved in Starting a Java Thread

- Creating & starting new threads on any Java platform consumes a non-trivial amount of system resources, so use them judiciously!
- e.g., only create threads for computations that run much longer than the time needed to spawn them!
End of Managing the Java Thread Lifecycle: Layers Involved in Starting a Thread